



UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE

United States Patent and Trademark Office

Address: COMMISSIONER FOR PATENTS

P.O. Box 1450

Alexandria, Virginia 22313-1450

www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
10/066,098	01/31/2002	Fabio Casati	10010118-1	6026
22879	7590	06/17/2011		
HEWLETT-PACKARD COMPANY Intellectual Property Administration 3404 E. Harmony Road Mail Stop 35 FORT COLLINS, CO 80528			EXAMINER NASH, LASHANYA RENEE	
			ART UNIT 2492	PAPER NUMBER
			NOTIFICATION DATE 06/17/2011	DELIVERY MODE ELECTRONIC

Please find below and/or attached an Office communication concerning this application or proceeding.

The time period for reply, if any, is set in the attached communication.

Notice of the Office communication was sent electronically on above-indicated "Notification Date" to the following e-mail address(es):

JERRY.SHORMA@HP.COM

ipa.mail@hp.com

laura.m.clark@hp.com



UNITED STATES PATENT AND TRADEMARK OFFICE

Commissioner for Patents
United States Patent and Trademark Office
P.O. Box 1450
Alexandria, VA 22313-1450
www.uspto.gov

**BEFORE THE BOARD OF PATENT APPEALS
AND INTERFERENCES**

Application Number: 10/066,098
Filing Date: January 31, 2002
Appellant(s): CASATI ET AL.

Dan C. Hu
For Appellant

EXAMINER'S ANSWER

This is in response to the appeal brief filed 24 March 2011 appealing from the Office action mailed 27 October 2010.

(1) Real Party in Interest

The examiner has no comment on the statement, or lack of statement, identifying by name the real party in interest in the brief.

(2) Related Appeals and Interferences

The examiner is not aware of any related appeals, interferences, or judicial proceedings which will directly affect or be directly affected by or have a bearing on the Board's decision in the pending appeal.

(3) Status of Claims

The following is a list of claims that are rejected and pending in the application:

Claims 1-3, 5-7, 11-14, 16-18, 27, 28, 30 and 31.

(4) Status of Amendments After Final

The examiner has no comment on the appellant's statement of the status of amendments after final rejection contained in the brief.

(5) Summary of Claimed Subject Matter

The examiner has no comment on the summary of claimed subject matter contained in the brief.

(6) Grounds of Rejection to be Reviewed on Appeal

The examiner has no comment on the appellant's statement of the grounds of rejection to be reviewed on appeal. Every ground of rejection set forth in the Office action from which the appeal is taken (as modified by any advisory actions) is being maintained by the examiner except for the grounds of rejection (if any) listed under the subheading "WITHDRAWN REJECTIONS." New grounds of rejection (if any) are provided under the subheading "NEW GROUNDS OF REJECTION."

(7) Claims Appendix

The examiner has no comment on the copy of the appealed claims contained in the Appendix to the appellant's brief.

(8) Evidence Relied Upon

The following is a listing of the evidence (e.g., patents, publications, Official Notice, and admitted prior art) relied upon in the rejection of claims under appeal.

Conversations + Interfaces = Business Logic	Kuno et al.	9-2001
---	-------------	--------

7,051,072	Stewart et al.	5-2006
-----------	----------------	--------

An architecture for a Secure Service Discovery Service

Czerwinski et al.	8-1999
-------------------	--------

(9) Grounds of Rejection

The following ground(s) of rejection are applicable to the appealed claims:

Claims 1, 11-14, 16-19, 27, 29 and 31 are rejected under 35 U.S.C. 103(a) as being unpatentable over Kuno et al. ["Conversations + Interferences = Business Logic"-retrieved from the Internet] in view of Stewart et al. (US Patent 7,051,072), hereinafter referred to as Kuno and Stewart respectively.

In reference to claim 1, Kuno discloses:

- A method for selecting a conversation logic at run-time for a workflow definition that includes at least one node with no hard-coded conversation logic (abstract; title page), the method comprising the steps of:
- a) maintaining a conversation logic repository that includes plural conversation logic (i.e. UDDI registry of conversation definitions; *3.2 Web Service Conversation Language*, page 5), wherein each of the plural conversation logic is external to the workflow definition (i.e. conversation is implemented differently than the workflow logic; *5 Related Work*, page 12, paragraph 2), wherein each of the plural conversation logic specifies a corresponding set of operations to be performed on a respective service (WSCL specification; *3.2 Web Service Conversation Language*, page 6);
- b) when executing the node with no hard-coded conversation logic (i.e. e-service client not hard-coded with conversation logic; *4.1 Client automation*, page 10), dynamically discovering, by a computer, a service associated with the node with no hard-coded conversation logic, wherein the discovered service is selected from among plural services (i.e. e-services discovery; *2. Approach*, page 1);
- c) selecting one of the plural conversation logic in the conversation logic repository based on the discovered service (i.e. conversation mapped to appropriate service; *4. Dynamic Conversation Controller for E-services*; paragraphs 1-3; pages 8-9); and

However, the reference fails to disclose dynamically plugging in the determined selected conversation logic into the node at run time in the computer, wherein the run

time is a time during which the node with no hard-coded conversation logic is being executed. Nonetheless, this was well known in the art at the time of the invention as further evidenced by Stewart. Therefore, it would have been obvious for one of ordinary skill in the art to accordingly modify the teachings of Kuno.

In an analogous art, Stewart discloses a mechanism for providing real-time conversations among business logic. Stewart discloses dynamically plugging in the determined selected conversation logic into the node at run time in the computer, wherein the run time is a time during which the node with no hard-coded conversation logic is being executed (i.e. logic plug-ins; column 19, line 19-column 20, line 54). One of ordinary skill in the art would have been so motivated to accordingly modify the teachings of Kuno so as to enable business makers to design and implement business rules that meet their specific needs (column 10, lines 20-31).

In reference to claim 11, Kuno discloses:

- A system for dynamically selecting a conversation logic at run-time for a workflow definition that includes at least one node with no hard-coded conversation logic (abstract; title page) comprising:
 - a) a workflow engine (*Related Work*, page 12, paragraph 2) for processing workflow definition (i.e. service definition; 3.3. *Web-Service Definition Language*; page 7);
 - b) a conversation logic repository that includes plural conversation logic (i.e. UDDI registry of conversation definitions; 3.2 *Web Service Conversation*

Language, page 5) and that is external to the workflow definition (i.e. conversation is implemented differently than the workflow logic; 5 *Related Work*, page 12, paragraph 2), wherein each of the plural conversation logic specifies a corresponding set of operations to be performed on a respective service (WSCL specification; 3.2 *Web Service Conversation Language*, page 6);

- an engine (i.e. conversation controller) configured to select one of the plural services for execution of the node with no hard-coded conversation logic, (4.1 *Client Automation*, paragraph 2; pages 10-11);
- c) a dynamic conversation logic selection mechanism configured to receive a service identifier that is associated with the selected service at run-time and based on the service identifier to select a conversation logic from the plural conversation logic for interacting with the selected service at run-time (i.e. conversation mapped to appropriate service; 4. *Dynamic Conversation Controller for E-services*; paragraphs 1-3; pages 8-9).

However, the reference fails to disclose wherein the dynamic conversation logic selection mechanism is configured to further dynamically plug in the selected conversation logic into the node at run-time, where the run-time node is a time during which the node with no hard-coded conversation logic is being executed. Nonetheless, this was well known in the art at the time of the invention as further evidenced by Stewart. Therefore, it would have been obvious for one of ordinary skill in the art to accordingly modify the teachings of Kuno.

In an analogous art, Stewart discloses a mechanism for providing real-time conversations among business logic. Stewart discloses wherein the dynamic conversation logic selection mechanism is configured to further dynamically plug in the selected conversation logic into the node at run-time, where the run-time node is a time during which the node with no hard-coded conversation logic is being executed (i.e. logic plug-ins; column 19, line 19-column 20, line 54). One of ordinary skill in the art would have been so motivated to accordingly modify the teachings of Kuno so as to enable business makers to design and implement business rules that meet their specific needs (column 10, lines 20-31).

In reference to claim 12, Kuno discloses the system of claim 11 further comprising: a source for services; wherein the source is configured to discover services based on a service selection rule; wherein the dynamic conversation logic selection mechanism (DCLSM) (i.e. dynamic conversation controller) selects appropriate conversation logic from the conversation logic repository based on a discovered service, (*4. Dynamic Conversation Controller for E-services*; pages 8-9).

In reference to claim 13, Kuno discloses the system of claim 12 wherein the source is one of a service broker, a service marketplace, and an e-services platform (i.e. e-service; *4.1 Client automation*, page 10).

In reference to claim 14, Kuno discloses, wherein only services that have a

conversation protocol compatible with one of the conversation logic available in the repository are considered for selection for execution of the node (; *4.1 Client automation*, page 10).

In reference to claim 16, Kuno discloses wherein a particular one of the plural conversation logic is for the exclusive use of a given one of the plural services (*4. Dynamic Conversation Controller for E-services*; paragraphs 1-3; pages 8-9).

In reference to claim 17, Kuno discloses wherein another of the plural conversation logic is shared by two or more of the plural services (*4. Dynamic Conversation Controller for E-services*; paragraphs 1-3; pages 8-9).

In reference to claim 18, Kuno discloses wherein the selected conversation logic is not defined in a workflow at process definition time (*5 Related Work*, page 12, paragraph 2)

In reference to claim 19, Kuno discloses wherein the dynamic conversation logic selection mechanism is configured to perform late binding of the selected conversation logic at run-time (*4.1. Client automation*, page 10).

In reference to claim 27, Kuno discloses wherein different ones of the plural conversation logic are compatible with different ones of the plural services, and wherein selecting one of the plural conversation logic comprises selecting a conversation logic

that is compatible with the discovered service (3.3. *Web-Service Definition Language (WSDL)*, page 7).

In reference to claim 29, Kuno discloses, wherein the dynamic conversation logic is configured to dynamically plug the selected conversation logic into the node (3.2 *Web Service Conversation Language*, paragraph 2; page 5).

In reference to claim 31, Kuno discloses wherein different ones of the plural conversation logic are compatible with different ones of the plural services, and wherein selecting one of the plural conversation logic comprises selecting a conversation logic that is compatible with the discovered service (3.3. *Web-Service Definition Language (WSDL)*, page 7).

Claims 2, 3, 5-7 and 28 and 30 are rejected under 35 U.S.C. 103(a) as being unpatentable over Kuno et al. ["Conversations + Interferences = Business Logic"-retrieved from the Internet] in view of Czerwinski et al. ["An Architecture for a Secure Service Discovery Service"-retrieved from ACM database] and Stewart et al. (US Patent 7,051,072), hereinafter referred to as Kuno and Czerwinski and Stewart respectively.

In reference to claim 3, Kuno discloses:

- A method for selecting a conversation logic at run-time (abstract; title page) comprising the steps of:

- maintaining a conversation logic repository that includes at least one conversation logic (i.e. UDDI registry of conversation definitions; 3.2 *Web Service Conversation Language*, page 5), wherein each of the plural conversation logic specifies a corresponding set of operations to be performed on a respective service (WSCL specification; 3.2 *Web Service Conversation Language*, page 6);
- at run-time, sending a service selection query to an electronic services platform or other service broker (i.e. e-service discovery; 2 *Approach*, page 2) ;
- receiving a returned service identifier (i.e. business service information in message; 3.1. *UDDI Registries*, page 4); and selecting a conversation logic from the conversation logic repository based on the returned service identifier (i.e. conversation mapped to appropriate service; 4. *Dynamic Conversation Controller for E-services*; paragraphs 1-3; pages 8-9).

However, the reference fails to disclose sending a selection query to an electronic service platform or other service broker, wherein the service selection query is for selecting a service from among plural services, and the returned service identifier corresponding to the selected service. Nonetheless, this was a well known feature in service discovery in the art at the time of the invention, as further evidenced by Czerwinski. Therefore, it would have been obvious for one of ordinary skill in the art at the time of the invention to accordingly modify the teachings of Kuno.

In an analogous art, Czerwinski discloses architecture for service discovery in a networked environment (*abstract*). Furthermore sending a selection query to an electronic service platform or other service broker, wherein the service selection query

is for selecting a service from among plural services (i.e. clients employ queries for location of services; *abstract*), and the returned service identifier corresponding to the selected service (i.e. unique service key; 6.1 *DNS and Globe*; page 32). One of ordinary skill in the art would have been so motivated to accordingly modify the teachings of Kuno so as to support end-users locating a particular network service out a plurality (i.e. hundreds of thousands) of accessible services discovered (Czerwinski ; *abstract*). However, the reference fails to disclose dynamically plugging in the determined selected conversation logic into the node at run time in the computer, wherein the run time is a time during which the node with no hard-coded conversation logic is being executed; and dynamically plugging in the selected conversation logic into the node with no hard-coded conversation logic. Nonetheless, this was well known in the art at the time of the invention as further evidenced by Stewart. Therefore, it would have been obvious for one of ordinary skill in the art to accordingly modify the teachings of Kuno.

In an analogous art, Stewart discloses a mechanism for providing real-time conversations among business logic. Stewart discloses dynamically plugging in the determined selected conversation logic into the node at run time in the computer, wherein the run time is a time during which the node with no hard-coded conversation logic is being executed; and dynamically plugging in the selected conversation logic into the node with no hard-coded conversation logic (i.e. logic plug-ins; column 19, line 19-column 20, line 54). One of ordinary skill in the art would have been so motivated to

accordingly modify the teachings of Kuno so as to enable business makers to design and implement business rules that meet their specific needs (column 10, lines 20-31).

In reference to claim 2, Kuno and Stewart disclose the method of claim 1 wherein the step of when executing the node with no hard-coded conversation logic, dynamically discovering a service associated with the node with no hard-coded conversation logic includes the steps of: determining a service based on a service selection rule; receiving a service reference; and wherein the step of determining a corresponding conversation logic in the conversation logic repository based on the discovered service further includes the step of using the service reference to determine a conversation logic for the determined service (Kuno; *4. Dynamic Conversation Controller for E-services*; paragraphs 1-3; pages 8-9). However, the reference fails to disclose sending a selection query to an electronic service platform or other service broker, wherein the service selection query is for selecting a service from among plural services, and the returned service identifier corresponding to the selected service. Nonetheless, this was a well known feature in service discovery in the art at the time of the invention, as further evidenced by Czerwinski. Therefore, it would have been obvious for one of ordinary skill in the art at the time of the invention to accordingly modify the teachings of Kuno and Stewart.

In an analogous art, Czerwinski discloses architecture for service discovery in a networked environment (*abstract*). Furthermore sending a selection query to an electronic service platform or other service broker, wherein the service selection query

is for selecting a service from among plural services (i.e. clients employ queries for location of services; *abstract*), and the returned service identifier corresponding to the selected service (i.e. unique service key; 6.1 *DNS and Globe*; page 32). One of ordinary skill in the art would have been so motivated to accordingly modify the teachings of Kuno and Stewart so as to support end-users locating a particular network service out a plurality (i.e. hundreds of thousands) of accessible services discovered (Czerwinski ; *abstract*).

In reference to claim 5, Kuno discloses the method of claim 3 wherein a particular one of the plural conversation logic is for the exclusive use of a given one of the plural services (4. *Dynamic Conversation Controller for E-services*; paragraphs 1-3; pages 8-9).

In reference to claim 6, Kuno discloses the method of claim 3 wherein another of the plural conversation logic is shared by two or more of the plural services (4. *Dynamic Conversation Controller for E-services*; paragraphs 1-3; pages 8-9).

In reference to claim 7, Kuno discloses the method of claim 3 wherein the conversation logic is not defined in a workflow at process definition time, the workflow defining a procedure that executes services (5 *Related Work*, page 12, paragraph 2)

In reference to claim 28, Kuno discloses wherein different ones of the plural conversation logic are compatible with different ones of the plural services, and wherein selecting one of the plural conversation logic comprises selecting a conversation logic that is compatible with the discovered service (3.3. *Web-Service Definition Language (WSDL)*, page 7).

In reference to claim 30, Kuno and Stewart fail to disclose the engine to select one of the plural services configured to submit a selection service query to an electronic platform to perform selection of the selected services from the plural services. Nonetheless, this was a well known feature in service discovery in the art at the time of the invention, as further evidenced by Czerwinski. Therefore, it would have been obvious for one of ordinary skill in the art at the time of the invention to accordingly modify the teachings of Kuno and Stewart.

In an analogous art, Czerwinski discloses architecture for service discovery in a networked environment (*abstract*). Furthermore submitting a selection service query to an electronic platform to perform selection of the services from the plural services (i.e. clients employ queries for location of services; *abstract*), and the returned service identifier corresponding to the selected service (i.e. unique service key; 6.1 *DNS and Globe*; page 32). One of ordinary skill in the art would have been so motivated to accordingly modify the teachings of Kuno and Stewart so as to support end-users locating a particular network service out a plurality (i.e. hundreds of thousands) of accessible services discovered (Czerwinski ; *abstract*).

(10) Response to Argument

In considering Appellant's arguments the following remarks are noted:

- (I) Appellant contends that Stewart provides no teaching or hint of dynamically plugging in any conversation logic into any node of the workflow template.
- (II) Appellant contends that Examiner appears to have equated the node with the e-service client of Kuno, and thus us not the node of the workflow definition.
- (III) Appellant contends that since no reason existed that would have prompted a person of ordinary skill in the art to combine the teachings of Kuno and Stewart, as discussed in connection with claim 1, a person of ordinary skill in the art would also not have been prompted to combine the teachings of Kuno, Czerwinski, and Stewart to achieve the subject matter of claim 3.

In considering (I), Appellant contends that Stewart provides no teaching or hint of dynamically plugging in any conversation logic into any node of the workflow template. Examiner respectfully disagrees. Examiner additionally notes that Appellant suggests the workflow template as disclosed by Stewart is considered to correspond to the workflow definition recited in claim 1. (See Appeal Brief, page 8). However, Examiner notes that Stewart is not cited to teach the workflow definition. Rather, Stewart is cited for teaching dynamically plugging in conversation logic into the node at run time in the

computer. Examiner asserts that Stewart expressly discloses logic plug-ins that are employed for adding unique functionality in invoking complex business rules, namely conversation logic, during a conversation between trading partners at run time (column 19, line 50-column 20, line 17). Stewart further discloses that the logic plug-ins are employed at plug-in points, or rather a node in the collaboration process, for allowing the communicating customers to introduce code for implementing additional features. Examiner asserts that the aforementioned node as taught by Stewart is at least implicitly taught to be the node of the workflow. This notion is further evidenced as Stewart also expressly discloses that workflow definitions are "dynamic, flexible, and configurable that it may be readily modified to reflect company's business model and to add new elements..." (column 11, lines 25-35). Therefore, Examiner asserts that Stewart provides at least the suggestion for combining dynamically plugging in conversation logic plug-in into a node, as taught by Stewart, into a node a workflow definition, as taught by Kuno, so as to yield the predictable result of a dynamically modified workflow definition.

Furthermore, Examiner asserts that the aforementioned argument attacks the Stewart reference individually. However one cannot show nonobviousness by attacking references individually where the rejections are based on combinations of references. See *In re Keller*, 642 F.2d 413, 208 USPQ 871 (CCPA 1981); *In re Merck & Co.*, 800 F.2d 1091, 231 USPQ 375 (Fed. Cir. 1986). Examiner asserts that the combination of Kuno and Stewart teaches, inter alia, dynamically plugging in the

determined selected conversation logic into the node (included as part of the workflow definition) at run time in the computer, as set forth in the rejections.

In considering (II), Appellant contends that Examiner appears to have equated the node with the e-service client of Kuno, and thus is not the node of the workflow definition, as in claim 1. Examiner respectfully disagrees. Examiner asserts that Kuno discloses workflows which implement the web-based service functionality using integrated processes (5. *Related Work*, paragraph 4; page 12, pages 12-13). Similarly, Appellant defines a workflow node as a step in the process of a service's implementation (See Appellant's Specification; page 10, paragraph 3). Therefore, it is evident that the workflows, as taught by Kuno, contain nodes that are the particular processes of the service functionality. Examiner additionally notes that the citation mentioned by Appellant (See Appeal Brief, page 9) is not intended to equate the node of a workflow definition with the e-service client, but rather to evidence that the conversation logic is not hard-coded into a computer employed at run-time for the execution of conversation in the e-service environment. Thus, Kuno evidently discloses a workflow definition including at least one node.

Subsequently, Examiner asserts that there would be a suggestion to incorporate the dynamic plugging in of conversation logic plug-in into a collaboration node, as taught by Stewart, directly into the known workflow definition node as taught Kuno. Therefore, it would have been obvious for one of ordinary skill in the art at the time of the invention to accordingly modify the teachings of Kuno with the teachings of Stewart.

The Examiner recognizes that obviousness may be established by combining or modifying the teachings of the prior art to produce the claimed invention where there is some teaching, suggestion, or motivation to do so found either in the references themselves or in the knowledge generally available to one of ordinary skill in the art. See *In re Fine*, 837 F.2d 1071, 5 USPQ2d 1596 (Fed. Cir. 1988), *In re Jones*, 958 F.2d 347, 21 USPQ2d 1941 (Fed. Cir. 1992), and *KSR International Co. v. Teleflex, Inc.*, 550 U.S. 398, 82 USPQ2d 1385 (2007). In this case, as specifically discussed in addressing (I) above, there is suggestion to combine the teachings of Kuno and Stewart. Stewart discloses that workflow definitions are "dynamic, flexible, and configurable that it may be readily modified to reflect company's business model and to add new elements..." (Stewart; column 11, lines 25-35). Thereby, one of ordinary skill in the art would have been motivated to enable business makers to design and implement business rules, namely workflow definitions, which meet their specific needs (Stewart; column 10, lines 20-31) as reflected in dynamic mechanisms such as particularly selected conversation logic.

In considering (III), Appellant contends that since no reason existed that would have prompted a person of ordinary skill in the art to combine the teachings of Kuno and Stewart, as discussed in connection with claim 1, a person of ordinary skill in the art would also not have been prompted to combine the teachings of Kuno, Czerwinski, and Stewart to achieve the subject matter of claim 3. Examiner respectfully disagrees. As addressed in discussing (II) above there is clear and evident suggestion to combine the

teachings of Kuno and Stewart. Specifically, Stewart discloses that workflow definitions are "dynamic, flexible, and configurable that it may be readily modified to reflect company's business model and to add new elements..." (Stewart; column 11, lines 25-35). Thereby, one of ordinary skill in the art would have been motivated to enable business makers to design and implement business rules, namely workflow definitions, which meet their specific needs (Stewart; column 10, lines 20-31) as reflected in dynamic mechanisms such as particularly selected conversation logic. Subsequently, Examiner emphatically asserts that a person of ordinary skill in the art would also have been prompted to combine the teachings of Kuno, Czerwinski, and Stewart to achieve the subject matter of claim 3.

(11) Related Proceeding(s) Appendix

No decision rendered by a court or the Board is identified by the examiner in the Related Appeals and Interferences section of this examiner's answer.

For the above reasons, it is believed that the rejections should be sustained.

Respectfully submitted,

/LaShanya R Nash/

Examiner, Art Unit 2492

Art Unit: 2492

Conferees:

/Aravind K Moorthy/

Primary Examiner, Art Unit 2492

/saleh najjar/

Supervisory Patent Examiner, Art Unit 2492